**1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1 Исходные данные**

Для написания и отладки программы использовался Qt Creator. В качестве языка программирования был выбран язык C++ и фреймворк Qt для создания графического интерфейса (GUI).

Компиляция проекта происходила с помощью компилятора MinGW 64, а запуск на ОС Windows 10.

Для эмуляции COM-портов использовался Virtual Serial Ports Emulator, задействующий архитектуру UART 16550.

**1.2 Параметры инициализации COM-порта**

Qt Framework предоставляет классы QSerialPort и QSerialPortInfo для работы с COM-портами.

Объект класса QSerialPort имеет следующие параметры:

* BaudRate – выходная частота бод-генератора или скорость передачи данных. Для UART 16550 один бод (baud) равен одному биту в секунду. Задается методом setBaudRate(). Возможные значения могут высчитываться по формуле:

Fout = Fin / (16 \* DL),

где Fin – входная частота;

DL – шестнадцатибитная константа, старшая и младшая части которой хранятся в двух регистрах UART (DLL и DLM).

Стандартные значения (Hz): 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

По умолчанию задана частота 9600 Hz.

* DataBits – количество битов в одном байте. Устанавливается с помощью метода setDataBits(). Может принимать значения от 5 до 8. По умолчанию размер байта равен 5 битам.
* Parity – паритет (четность). Этот параметр нужен для проверки целостности информационной части. Паритет может принимать следующие значения, определенные в классе QSerialPort: NoParity (без четности), OddParity (проверка на нечетность единиц в байте, EvenParity (проверка на четность единиц в байте), MarkParity (проверка на четность/нечетность отсутствует, но бит паритета равен единице), SpaceParity (проверка на четность/нечетность отсутствует, но бит паритета равен нулю). При OddParity бит паритета устанавливается в 0 или 1 в зависимости от количества нечетных единиц в байте, при EvenParity – в зависимости от количества четных единиц. По умолчанию – NoParity.
* StopBits – стоп-биты. По умолчанию линия находится в состоянии логической единицы. При наличии байта для передачи передатчик переводит линию в состояние логического нуля, т. е. передает старт-бит, что говорит приемнику о том, что на следующем такте нужно “ловить” первый информационный бит. Стоп-бит необходим для того, чтобы после передачи информационной последовательности гарантированно вернуть линию в исходное, т.е. единичное состояние. Старт-бит всегда один, а стоп-битов может быть 1, 1.5 либо 2. По умолчанию устанавливается 1 стоп-бит.
* FlowControl – управление потоком. Этот параметр, к примеру, позволяет избегать переполнения приемника, приостанавливая “быстрый” передатчик.

Возможные значения: NoFlowControl (без управления потоком), HardwareControl (аппаратное управление потоком), SoftwateControl (программное управление потоком). По умолчанию устанавливается в NoFlowControl.

* PortName – имя порта. Примеры: “COM1”, “COM2”. При старте программы устанавливается первый найденный свободный открытый порт..
* ReadBufferSize – размер буфера чтения. Задается в байтах (целое число).
* ReadDataMode – режим чтения данных. Возможные значения для инициализации: AllDataMode (чтение всех доступных данных), ReadLineMode (чтение до символа новой строки), ReadPortUnbuffered (не буферизованное чтение). AllDataMode – значение по умолчанию.
* WriteBufferSize – размер буфера записи. Задается в байтах (целое число).

**2 ПРАКТИЧЕСКАЯ ЧАСТЬ**

**2.1 Код программы**

**………………………………………………….…**

**…………………………………………………….**

**…………………………………………………….**

**2.2 Интерфейс программы**

Интерфейс программы представлен на рисунке 2.1.

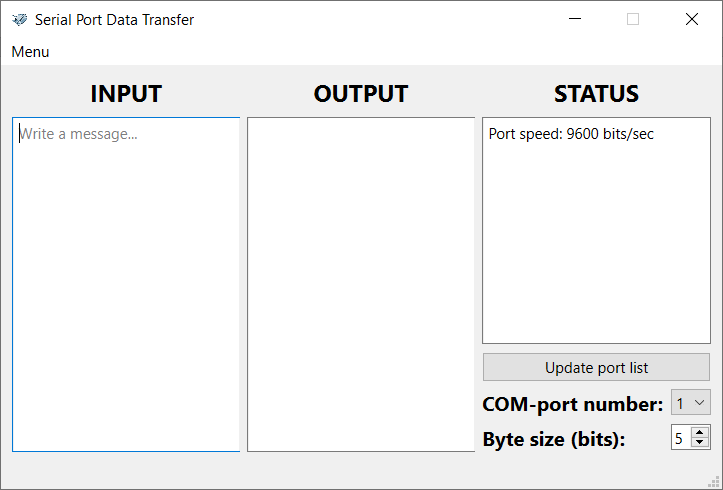


Рисунок 2.1 – Интерфейс программы